An Autonomic Reliability Improvement System for Cyber-Physical Systems

Leon Wu Gail Kaiser
Department of Computer Science
Columbia University
New York, USA
{*leon,kaiser*}*@cs.columbia.edu*

***Abstract*—System reliability is a fundamental requirement of cyber-physical systems. Unreliable systems can lead to disruption of service, financial cost and even loss of human life. Typical cyber-physical systems are designed to process large amounts of data, employ software as a system component, run online continuously and retain an operator-in-the-loop because of human judgment and accountability requirements for safety-critical systems. This paper describes a data-centric runtime monitoring system named ARIS *(Autonomic Reliability Improvement System)* for improving the reliability of these types of cyber-physical systems. ARIS employs automated online evaluation, working in parallel with the cyber-physical system to continuously conduct automated evaluation at multiple stages in the system workflow and provide real-time feedback for reliability improvement. This approach enables effective evaluation of data from cyber-physical systems. For example, abnormal input and output data can be detected and flagged through data quality analysis. As a result, alerts can be sent to the operator-in-the-loop, who can then take actions and make changes to the system based on these alerts in order to achieve minimal system downtime and higher system reliability. We have implemented ARIS in a large commercial building cyber-physical system in New York City, and our experiment has shown that it is effective and efficient in improving building system reliability.**

***Keywords*-cyber-physical system; system reliability; reliability engineering; data analysis; machine learning; data mining; runtime environment; smart buildings**

I. INTRODUCTION

System reliability is a fundamental requirement of cyber-physical systems–*i.e.*, systems featuring a tight combination of and coordination between computational systems and physical elements. These include systems that manage critical infrastructure for essential functions ranging from power grids and transportation systems to biomedical instruments and devices. Unreliable systems can result in disruption of service, financial cost and in some cases even loss of human life [1]. More importantly, cyber-physical systems cannot be deployed for certain mission-critical applications such as traffic control, automotive safety or healthcare without improved reliability and predictability [2].

Typical cyber-physical systems are designed to meet the following criteria: process large amount of data; employ software as a system component; run online continuously; maintain an operator-in-the-loop because of human judgment and accountability requirements for safety-critical systems [3]. Systems that meet these criteria include building systems, power grids, energy systems, transportation systems, defense systems, factory automation systems and cloud computing data centers. These systems do not operate in a controlled environment, and must be robust to unexpected conditions and adaptable to subsystem failures [3]. It is often not possible to perform robust testing of cyber-physical systems prior to actual deployment because the physical devices are so expensive that they cannot be replicated in the testing lab, or at least not for large-scale operation. Thus, it is imperative to have an online quality assurance process that can continuously evaluate the live system during runtime in the field to ensure that it is performing reliably and as expected.

This paper describes a data-centric runtime monitoring platform named ARIS *(Autonomic Reliability Improvement System)*. ARIS works in parallel with the cyber-physical system, continuously conducting automated online evaluation at multiple stages along the system workflow and providing operator-in-the-loop feedback for reliability improvement.

One technique employed by ARIS is *data quality analysis*, wherein computational intelligence is applied to evaluate data quality in an automated and efficient way. ARIS also makes use of *self-tuning*, automatically self-managing and self-configuring the evaluation system to ensure that it adapts itself to both changes in the system and feedback from the operator. This self-tuning continuously adapts the evaluation system to ensure proper function, which leads to a more robust evaluation system and improved system reliability.

In the following section, we describe our approach of automated online evaluation, followed by the system architecture in section III. In section IV, we describe our empirical study. Finally, we compare some related work in section V before concluding in section VI.

II. APPROACH

As illustrated in Figure 1, automated online evaluation works in parallel with the cyber-physical system to perform continuous assessment at multiple stages along the system workflow and provide operator-in-the-loop feedback for reliability improvement. This enables ongoing evaluation of data from cyber-physical systems. For example, abnormal input and output data can be detected and flagged based

on data quality analysis. As a result, alerts can be sent out that enable the operator-in-the-loop to take actions and make changes to the system in order to minimize system downtime and maximize system reliability.

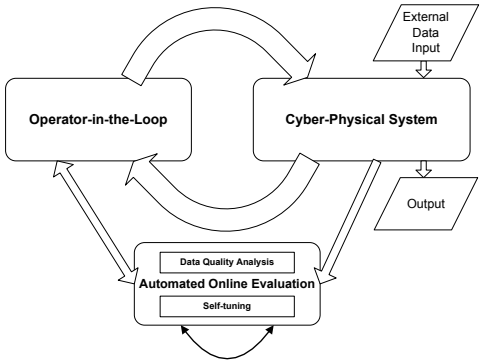

Figure 1. Automated online evaluation.

A. Data Quality Analysis

ARIS uses computational intelligence to perform data quality analysis in an automated and efficient way and thereby ensure that the running system performs as reliably as possible. This computational intelligence is enabled by machine learning, data mining, statistical and probabilistic analysis and other intelligent techniques. In a cyber-physical system, data collected from the system (*e.g.*, sensor data points, software bug reports, system status logs and error reports) are stored in databases. ARIS analyzes these data so that useful information on system reliability, such as erroneous data or abnormal system states, can be obtained. This reliability-related information is in turn directed to system operators so that proper actions can be taken–in some cases, proactively based on predictive results–to ensure proper and reliable execution of the system. The following are some data quality analysis techniques used by ARIS.

1) Thresholds: The thresholds define the normal working range for specific data-points. If data readings exceed these thresholds at either the lower or upper bound, the data record will be flagged as anomalous and a corresponding warning will be communicated back to the operator electronically.

2) Online Anomaly Detection for Single Data Points: Anomaly detection is used to find data instances that are unusual and do not fit any established pattern. It concentrates on modeling normal behavior in order to identify atypical data-points. For the cyber-physical systems of interest in this study, time-series data usually arrive continuously in parallel at a varied pace. This component processes the continuously updated data-streams to detect anomalies for single data-points, using a customized incremental Local Outlier Factor (LOF) algorithm [4]. The algorithm uses k-nearest neighbor on each inserted data record to instantly compute LOF value, which is the degree to which a data record represents an outlier or an indicator of abnormality. A sudden increase in LOF value indicates that a data record is likely to be an outlier. LOF values for existing data records can be updated on the fly if necessary. Because each data series is for an individual data source with low data dimensionality, such as a sensor's reading of (time, value) tuples, the incremental LOF algorithm is computationally efficient.

Furthermore, multiple LOF value time series can be processed for different data sources and displayed in parallel via *sparkline graph*, a type of information graphic characterized by its small size and high data density [5], for more fine-grained checks. As shown in Figure 2, data series B and C both experience a sudden increase of LOF value at around the 38th hour after the start of observation. This spike indicates a strong likelihood of a data anomaly at that time-point. This visualization component provides an easy way to obtain additional verification of a data anomaly. It is also a useful communication channel to help the operator understand where issues are arising.

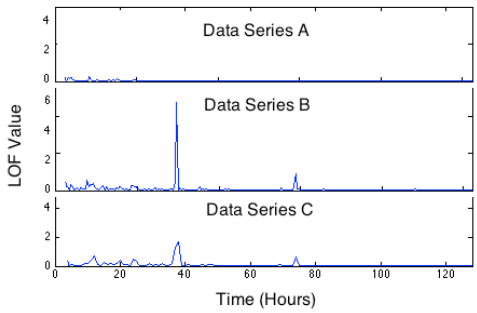

Figure 2. LOF value time series displayed using sparkline graph.

3) Data Anomaly Diagnosis Using Machine Learning Classification: After a data anomaly is detected, further automated diagnosis or reasoning is needed to infer what physical and computational/software component the data anomaly relates to, what reliability issue this anomaly might cause, and the recommended action (or work order) for the operator to take in order to correct the problem. This can be solved as a supervised learning problem and a classification model trained on existing data can predict unknown values (*i.e.*, the component having issues and the corresponding corrective/preventive action).

We used Support Vector Machines (SVMs) [6], [7] as the classifier. SVMs formulate the classification modeling process as a quadratic minimization problem and find hyperplanes in a high-dimensional space that separate data instances of different categories while maximizing the margins between categories. First, a set of historic data records (*e.g.*, each one with N attributes) is used as training data to build a linear SVM model as a classifier. For a new data record with one unknown data field A (*i.e.*, $N-1$ attributes available and one attribute or class label unknown),

the trained SVM model and the available $N-1$ attributes are used to predict the value of the unknown A field for this data record. In cases where multiple data fields need to be determined for a data record (*e.g.*, $N-M$ attributes available and M attributes unknown), the SVM model and the available $N-M$ available attributes are used to predict the unknown fields one by one.

Using SVM classification as the basis for data anomaly diagnosis has some advantages over rule-based reasoning systems. SVM classification does not require a lot of prior knowledge of the system because it works solely based on the data itself. Rule-based systems require derivation of the rules, including both forwarding-chaining rules (*e.g.*, IF something happens THEN do something) and backward-chaining rules (*e.g.*, IF I want to achieve this goal THEN something has to happen), based on extensive heuristics and in many cases expert domain knowledge. Also, SVM classification is adaptive based on updated training data, while rule-based logics are often rigid and not easy to change. In some unexpected real-world situations, rule-based systems are often unable to reach any conclusion whereas machine-learning approaches may be able to derive partially useful information for the operator, such as a rank list with scores based on probability and susceptibility.

B. Self-Tuning

ARIS also makes use of *self-tuning*, an aspect of autonomic computing [8] that automatically self-manages and self-configures the evaluation system to ensure that it adapts itself to changes in the system and feedback from the operator. This self-tuning is used to improve accuracy, efficiency and robustness of data analysis, and also minimizes the burden imposed on the operator.

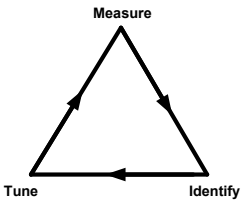

Figure 3. MIT self-tuning process.

As illustrated in Figure 3, self-tuning employs a Measure-Identify-Tune (MIT) process in order to achieve the following results:

- use performance metrics such as R^2 (coefficient of determination), ROC (receiver operating characteristic) and AUC (area under the curve) to measure and improve accuracy of the data analysis models [9]
- use statistical trend detection and curve fitting, such as Weibull distribution and parameters estimation [10], [11], to reduce variability and eliminate overshoot
- prioritize updates from operators and adjust system parameters such as set-points, thresholds and machine learning model parameters when abnormal exogenous situations happen in order to reduce false alarms
- use dynamic load balancing and failover switch, which applies to the parallel processing of the large amount of time series data coming from different data sources, to maximize efficiency and reliability

III. ARCHITECTURE

The architecture of ARIS is illustrated as a seven-step process in Figure 4. ARIS evaluates the cyber-physical system via three stages of data quality analysis (steps 13): first, evaluation of the input data; second, evaluation of the data output; and third, evaluation of feedback from the cyber-physical system.

The initial evaluation checks to see if the input data meets the quality specifications pre-defined by the application developer and the system operator. Examples of data quality specification include data existence, up-to-date, conforming to certain distribution, time-synchronization across different sources, variation and pattern.

The output data evaluation checks the quality of the results of the application. For example, for a machine learning-based prediction system, data output quality relates to the accuracy or confidence level of the prediction. For a non-machine learning-based system, such as a building energy management system, the quality of the data output relates to the extent to which results can be used to guide subsequent actions (*e.g.*, building energy use adjustment).

The evaluation of the feedback from the cyber-physical system checks the outcome resulting from the previous steps. This evaluation is important to ensure that the data output in fact leads to the desired system outcome.

In step 4, the results from the data quality analysis are directed to a user interface for system operators, who may take control or recovery actions when abnormal and erroneous situations happen. These actions ensure proper execution of the system and lead to improved system reliability.

At steps 5 and 6, the self-tuning component receives feedback from both the operator-in-the-loop and changes in the system.

Finally, in step 7, the self-tuning component self-manages and self-configures the evaluation system based on the feedback from the operator and the changes in the system. This self-tuning adapts the evaluation system to ensure proper functioning, which leads to a more robust evaluation system and improved system reliability.

To further illustrate the proposed architecture, here is an example use case wherein multiple steps and actions were managed using ARIS. A *Building Management System (BMS)* is a type of cyber-physical system consisting of both software and hardware components that controls and monitors a building's mechanical and electrical equipment, such as ventilation, lighting, power systems, fire systems and security systems. The building energy control system

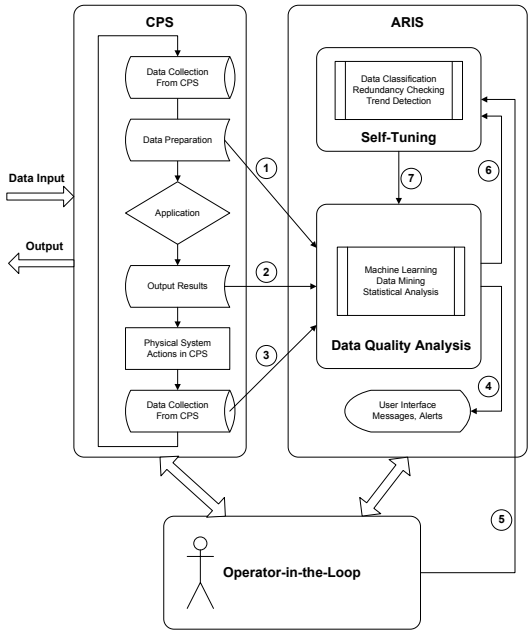

Figure 4. ARIS system architecture.

is an important component of the BMS that reads data feeds representing internal and exogenous conditions (*e.g.*, temperature, humidity, electrical load, peak load, fluctuating electricity pricing and building work schedule) and takes control actions (*e.g.*, adjust lighting, turn on/off the air-conditioning and shut off partial elevators) accordingly. Building operators usually have the ability to change or override control actions taken by the BMS to accommodate special situations such as severe weather or changes in the building's work schedule.

To ensure that the building energy control system works reliably 24x7, we evaluated input data, output data (*i.e.*, control actions) and the result of actions taken using ARIS. In one example scenario, a malfunction of the digital thermostat caused a temperature reading to stay at a fixed level without changing for a long time. The building energy control system was designed to accept any value within a certain temperature range and would not be able to handle this type of input data error (*i.e.*, constant temperature). In contrast, ARIS's intelligent data quality analysis component can quickly detect this type of input data error (Figure 4, step 1), and give feedback to the building operator (Figure 4, step 4). After receiving an automated notification from ARIS, the building's operator can then take appropriate action.

In another example scenario, building management notifies the operator of the need to keep the building fully functioning for a special, one-time-only event during the coming weekend. The operator then notifies ARIS about the abrupt change (Figure 4, step 5). The self-tuning component of the ARIS takes this signal and uses it to adjust data quality analysis (Figure 4, steps 6 and 7), thus avoiding possible false-positive system warnings due to the abnormal energy use data during this specific weekend.

IV. Empirical Study

A. Implementation

We have developed a prototype ARIS application. As shown in Figure 5, the software consists of a secure IP-based data connector, a data quality analysis and self-tuning module with back-end database, several feedback mechanisms (including alert emails, warning messages, and reports,) and a user interface enhanced by real-time visualization.

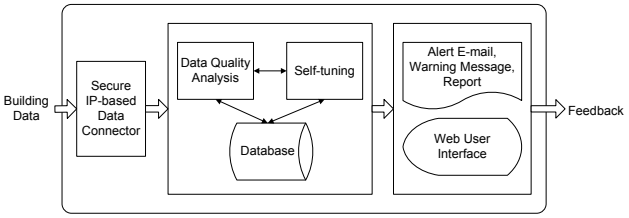

Figure 5. Software components and data flow.

B. Real-World Experiment

We implemented the prototype ARIS application at 345 Park Avenue, a 634 ft (193 m) tall skyscraper in midtown Manhattan, New York City. Designed by Emery Roth & Sons and completed in 1969, the building has 44 floors and more than 2 million square feet of tenant space. Approximately 5,000 people work in the building, and there are about 1,000 visitors to the building daily. Rudin Management, one of the largest private real estate companies in New York City, is the building owner and property manager. Building management has installed a state-of-the-art building energy monitoring system and BMS, which provided a live building dataset for ARIS.

As shown in Figure 6, ARIS worked with the BMS in parallel and processed the live data feeds via a remote data link. In our experiments, we connected ARIS to the building's various intelligent systems directly using a secure IP-based data connector. This setup simplifies the data collection and communication processes.

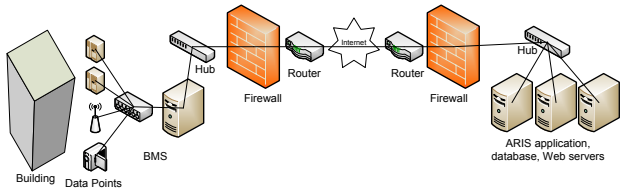

Figure 6. Experimental setup.

C. Experimental Results

Our experiments showed that ARIS is effective in ensuring that building systems operate reliably online continuously and in real-time. ARIS efficiently identified a large number of suspicious data anomalies obtained from 2,480 building data sources, mostly sensors, over a six-month period (December 2011 to May 2012). We investigated the relevant sensors and SCADA (supervisory control and data acquisition) data sources with the building operators and engineers. The results confirmed that the majority of the issues identified were in fact caused by system failures such as BMS software errors or equipment malfunctions. Figures 7-9 present some example time-series visualization charts for selected data sources.

Figure 7 shows out-of-bounds supply air temperatures that are lower than 50 °F or higher than 80 °F. After these abnormal behaviors are detected and flagged by ARIS, the building's operator can take proper control actions to maintain normal operation of the building's cooling system and ensure that service will not be disrupted. Through its SVM classification-based diagnosis, ARIS also recommends corrective actions to the operator as part of a work order management system.

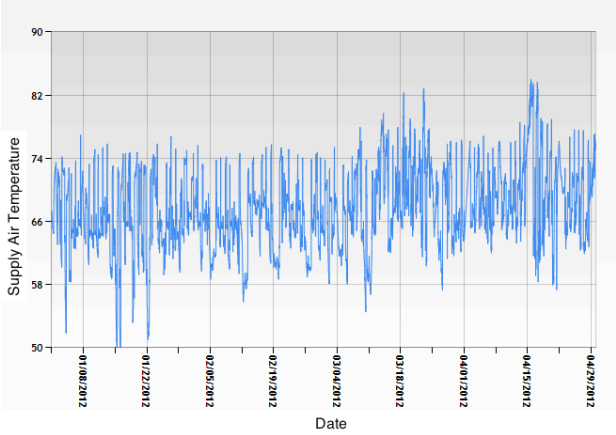

Figure 7. Supply air temperature time series.

As shown in Figures 8-9, the drop in maximum energy demand and steam demand around January 1 coincide with the building system shutdown during New Year's Eve and subsequent reactivation after the holiday. This kind of dip would normally be detected as anomalous behavior and a warning would be triggered and sent to building management from the automated online evaluator. However, the self-tuning capability allows the building's operator to notify ARIS about this abrupt schedule change to avoid the generation of unnecessary warnings.

V. Related Work

Reliability is widely recognized as a critical requirement for cyber-physical systems. In his paper "Cyber physical

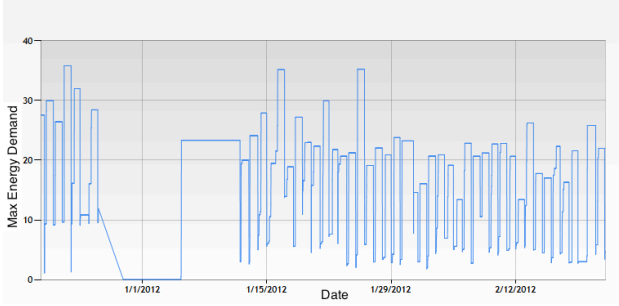

Figure 8. Maximum energy demand time series.

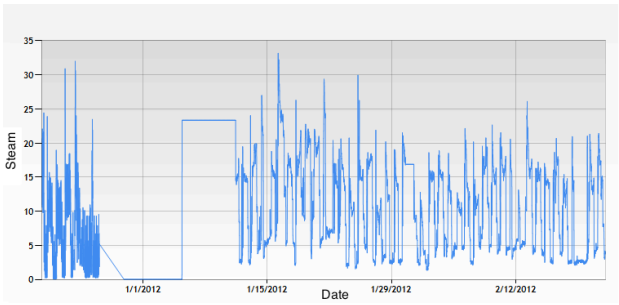

Figure 9. Maximum steam demand time series.

systems: Design challenges", Lee described how the expectation of reliability in cyber-physical system will only increase, and concluded that cyber-physical system will not be deployed into certain mission-critical applications such as traffic control, automotive safety and health care without improved reliability and predictability [2]. The CPS steering group stated in the executive summary of its 2008 Summit report that architectures and tools are needed to build reliable and resilient cyber-physical systems [3].

Some prior research has been done on data quality analysis and self-tuning. Gegick *et al.* performed text mining of bug reports to identify security issues [12]. Sullivan demonstrated that probabilistic reasoning and decision-making techniques can be used as the foundation for an effective, automated approach to software tuning [13]. Kaiser *et al.* have retrofitted autonomic computing onto legacy systems externally, without any need to understand or modify the code and, in many cases, even when it is impossible to recompile [14], [15].

Some prior research has also been done on smart building systems. A measurement and actuation profile for building information based on sensor systems was discussed in Ref. [16]. Their work is complementary to our approach. Schein and Bushby developed a rule-based system-level fault detection and diagnostic method for HVAC systems [17]. As described in section II-A3 above, the machine learning-based approach has certain advantages over rule-based systems.

VI. Conclusion

This paper presents a data-centric runtime monitoring system named ARIS that performs data quality analysis using computational intelligence and self-tuning techniques to improve system reliability for cyber-physical systems that process large amounts of data, employ software as a system component, run online continuously and maintain an operator-in-the-loop. Our experiments with ARIS in a large commercial building in New York City have demonstrated that this approach is effective and efficient. The data-dependence of this system makes it easily applicable to different types of cyber-physical systems, and the open expandable architecture also enables the incorporation of new data quality analysis and self-tuning techniques.

Acknowledgment

The authors would like to thank Dr. Christian Murphy and Dr. Roger Anderson for their comments, and Rudin Management Company for providing the experimental environment for this research. Wu and Kaiser are members of the Programming Systems Laboratory, funded in part by NSF CCF-1161079, NSF CNS-0905246, and NIH 2 U54 CA121852-06. Wu is also a member of the Energy Research Group in the Center for Computational Learning Systems, supported in part by General Electric, FedEx, Consolidated Edison, and Rudin Management Company.

References

[1] S. M. Amin, "U.S. electrical grid gets less reliable," *IEEE Spectrum*, p. 80, January 2011.

[2] E. A. Lee, "Cyber physical systems: Design challenges," in *International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC)*, May 2008, invited Paper. [Online]. Available: http://chess.eecs.berkeley.edu/pubs/427.html

[3] CPS Steering Group, "Cyber-physical systems executive summary," in *CPS Summit 2008*, March 2008, http://varma.ece.cmu.edu/Summit/.

[4] D. Pokrajac, A. Lazarevic, and L. J. Latecki, "Incremental local outlier detection for data streams," in *Proceedings of IEEE Symposium on Computational Intelligence and Data Mining*, 2007, pp. 504–515.

[5] E. Tufte, *Beautiful Evidence*. Graphics Press, 2006.

[6] V. N. Vapnik, *The nature of statistical learning theory*. New York: Springer-Verlag, 1995.

[7] C. Cortes and V. Vapnik, "Support-vector networks," in *Machine Learning*. Springer, 1995, p. 20.

[8] IBM, "Autonomic computing," 2011, available at http://www.research.ibm.com/autonomic/.

[9] L. Wu, G. Kaiser, C. Rudin, and R. Anderson, "Data quality assurance and performance measurement of data mining for preventive maintenance of power grid," in *Proceedings of the 17th ACM SIGKDD Workshop on Data Mining for Service and Maintenance*, August 2011.

[10] L. Wu, B. Xie, G. Kaiser, and R. Passonneau, "BugMiner: Software reliability analysis via data mining of bug reports," in *Proceedings of the 23th International Conference on Software Engineering and Knowledge Engineering (SEKE)*, July 2011.

[11] S. E. Rigdon and A. P. Basu, "Estimating the intensity function of a Weibull process at the current time: Failure truncated case," in *Journal of Statistical Computation and Simulation (JSCS)*, vol. 30, 1988, pp. 17–38.

[12] M. Gegick, P. Rotella, and T. Xie, "Identifying security bug reports via text mining: An industrial case study," in *Proceedings of the 7th IEEE Working Conference on Mining Software Repositories (MSR)*, Cape Town, May 2010, pp. 11–20.

[13] D. G. Sullivan, "Using probabilistic reasoning to automate software tuning," Harvard University, Tech. Rep., September 2003.

[14] G. Kaiser, "Autonomizing legacy systems," in *2002 IBM Almaden Institute Symposium on Autonomic Computing*, April 2001.

[15] J. Parekh, G. Kaiser, P. Gross, and G. Valetto, "Retrofitting autonomic capabilities onto legacy systems," *Journal of Cluster Computing*, vol. 9, no. 2, pp. 141–159, April 2006.

[16] S. Dawson-Haggerty, X. Jiang, G. Tolle, J. Ortiz, and D. Culler, "sMAP – a simple measurement and actuation profile for physical information," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys'10)*, November 2010.

[17] J. Schein and S. T. Bushby, "A hierarchical rule-based fault detection and diagnostic method for HVAC systems," *HVAC&R Research*, vol. 12, no. 1, January 2006.